WebDAQ Raspbian Usage and Installation

The MCC WebDAQ is based on the Raspberry Pi Compute Module 3 but runs a customized version of embedded Linux when configured by MCC. This document describes how to install the Raspbian operating system and use the WebDAQ as a Linux computer with data acquisition capability. Following these instructions will overwrite your WebDAQ system software and any settings or data stored on the internal media. The system software may be installed again by obtaining a factory install image from MCC and using it as the image in the installation instructions below.

Hardware Overview

- Raspberry Pi Compute Module 3
    - 4GB eMMC flash
    - Quad-core processor
    - 1GB RAM
- SD card connector
    - Uses SDIO interface on Compute Module
    - Uses mmc1 device in Raspbian
    - User must manually mount the device or create a udev rule to automatically mount it.
- Ethernet interface
    - LAN9514 USB 2.0 Hub / 10/100 Ethernet interface
    - Automatically supported by Raspbian
- Dual USB 2.0 connectors
    - User must manually mount flash drives or create a udev rule to automatically mount them.
- Data acquisition interface
    - USB-9162
    - Currently supports 9213 and 9234 modules
    - Can be used with C and Python code using the MCC driver and library
- Digital I/O
    - 4 bidirectional terminals
    - Electrically isolated from the rest of the board
    - Open drain outputs with fixed pullups to 5V
    - Input signals read the state of the DIO terminal:
        - DIO0: GPIO36
        - DIO1: GPIO37
        - DIO2: GPIO38
        - DIO3: GPIO39
    - Output signals control the output MOSFET (write 1 to drive DIO terminal low, 0 to turn MOSFET off and allow terminal to pull high):
        - DIO0: GPIO40
        - DIO1: GPIO41
        - DIO2: GPIO42

        - DIO3: GPIO43
- Buttons
    - Soft power on / shutdown button
        - Jumper W1 disables the power on/off function, may be used as a general-purpose button.
        - Sends the KEY_POWER keyboard event
        - Event can be overridden with power_button_keycode device tree parameter.
    - User button
        - Sends the KEY_A keyboard event
        - Event can be overridden with user_button_keycode device tree parameter.
    - Recessed network settings reset button
        - Sends the KEY_B keyboard event
        - Event can be overridden with reset_button_keycode device tree parameter.
- LEDs
    - Four bi-color green/yellow LEDs visible externally
        - Each LED is controlled by the green and yellow signals. One side must be high and the other low for the LED to turn on. For example:

power_led_green	power_led_yellow	Power LED State
0	0	Off
0	1	On Yellow
1	0	On Green
1	1	Off

        - LEDs can be controlled with sysfs under /sys/class/leds
            - power_led_green
            - power_led_yellow
            - status_led_green
            - status_led_yellow
            - error_led_green
            - error_led_yellow
            - media_led_green
            - media_led_yellow
        - Device tree trigger parameters can be used in /boot/config.txt to set up OS triggers to control the LEDs:
            - power_led_green_trigger
            - power_led_yellow_trigger
            - status_led_green_trigger
            - status_led_yellow_trigger
            - error_led_green_trigger
            - error_led_yellow_trigger
            - media_led_green_trigger
            - media_led_yellow_trigger
    - Two green surface mount LEDs for Compute Module
        - DS3 defaults to mmc0 (eMMC) activity.

            - Override with act_led_trigger device tree parameter in /boot/config.txt.
            - May be controlled with sysfs using /sys/class/leds/led0
        - DS4 defaults to mmc1 (SD) activity.
            - Override with led1_trigger device tree parameter in /boot/config.txt.
            - May be controlled with sysfs using /sys/class/leds/led1
    - USB-9162 green surface mount LED
        - Compute module can read the status of this LED with GPIO5.
- Real time clock with battery backup
    - DS1374U
    - Accessible in Raspbian as /sys/bus/i2c/devices/1-0068
- Programming connector
    - Internal micro USB connector
    - Allows overwriting the Compute Module eMMC with a Raspbian image
    - Same functionality as Compute Module development board
- 10-pin header

Pin	Function
1	+3.3V
2	GPIO8
3	GPIO9
4	GPIO10
5	GPIO11 (controls DS4 internal LED)
6	GPIO12
7	GPIO13
8	GPIO14 (UART Tx)
9	GPIO15 (UART Rx)
10	Ground

Installing Raspbian (Windows instructions)

1. Remove WebDAQ from enclosure.
2. Remove the data acquisition module from the WebDAQ board, being careful not to break the narrow part of the board.
3. Plug a micro usb cable into programming connector then insert the cable into a USB port on your Windows computer.
4. Install RPi-boot from https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot_setup.exe.
5. Power on WebDAQ, then run RPi-boot. It should detect the CM3 and download a small image that makes it appear as a new USB flash drive. This maps the onboard flash memory so it can be written.
6. Download a Raspbian Lite image from https://www.raspberrypi.org/downloads/raspbian/. Standard Raspbian contains many extra graphical interface packages that are not needed with the WebDAQ and may not fit on the eMMC.

7. Use a tool such as Etcher (https://etcher.io/) to write the Raspbian Lite image to the CM3 USB flash drive.
8. After the write has completed open the CM3 USB flash drive in a file explorer. You should see the Raspbian boot folder files (you may need to refresh your file explorer or unplug the USB cable from your PC and plug it back in.)
9. Download the files config.txt, dt-blob.bin, and webdaq.dtbo from https://github.com/nwright-mcc/webdaq_raspbian.
    a. https://github.com/nwright-mcc/webdaq_raspbian/raw/master/dt-blob.bin
    b. https://github.com/nwright-mcc/webdaq_raspbian/raw/master/webdaq.dtbo
    c. https://github.com/nwright-mcc/webdaq_raspbian/raw/master/config.txt
10. Copy config.txt and dt-blob.bin to the root of the USB drive.
11. Copy webdaq.dtbo to the **overlays** directory of the USB drive.
12. If you wish to have ssh support immediately, create an empty file in the root of the USB drive named "ssh". Raspbian will detect this file and enable ssh.
13. Do a safe removal of the USB drive.
14. Power off the WebDAQ and **unplug the USB cable from the programming connector**.
15. If you wish to connect to Raspbian using a serial terminal:
    a. Connect a serial port cable to the 10-pin header
        i. You need a serial cable with TTL interface, or an RS-232 to TTL adapter.
        ii. Connect ground from the cable to WebDAQ ground (pin 10.)
        iii. Connect Tx from the cable to WebDAQ Rx (pin 9.)
        iv. Connect Rx from the cable to WebDAQ Tx (pin 8.)
        v. If the cable requires an external supply, connect it to WebDAQ 3.3V (pin 1.) **Warning**: Many cables have a supply output, do not connect this to the WebDAQ.
    b. Open a terminal with an application such as Putty (https://www.putty.org/).
    c. Configure the serial port for 115,200 baud, 8 data bits, no parity bits, 1 stop bit.
    d. Connect to the serial port in the terminal program. Any boot messages should appear when you power on the WebDAQ, and you can log in through this terminal.
16. Power on the WebDAQ. The internal LED DS3 will be the activity LED and display eMMC activity.
17. You can log in to Raspbian normally at the command prompt in the serial terminal (user **pi**, password **raspberry**) or via ssh if you enabled it above. You will need to know the WebDAQ IP address to use ssh. If your PC supports the Bonjour protocol then you can connect with the name "raspberrypi.local".
18. Use the command "sudo raspi-config" to set the locale settings, change the device name and password, and configure other Raspbian settings.
19. Once you have verified proper Raspbian operation, power off the WebDAQ and re-attach the data acquisition module using the original hardware.

Set up power button control

The power button is used as a soft power on and shutdown button by default. The W1 jumper must be in place for this to work. The power button is set up to return the KEY_POWER (116) keyboard code, which initiates a shutdown in Raspbian when received.

The power may be configured to be always-on by removing W1. If W1 is removed, the power button may be used as a general-purpose button by changing the keyboard code to a different event. This is done by uncommenting the power_button_keycode in /boot/config.txt and setting the desired keyboard code

```
dtparam=power_button_keycode=46
```

The “46” above is the keyboard code to return; in this example, it is KEY_C.

Changing LED behavior

The LEDs are mapped to sysfs files so they may be turned on / off or be set to various system triggers. The behavior can be modified by uncommenting the desired line in /boot/config.txt and setting the trigger.

```
dtparam=status_led_green_trigger=”mmc0”
```